

PRE-ENTREGA 2

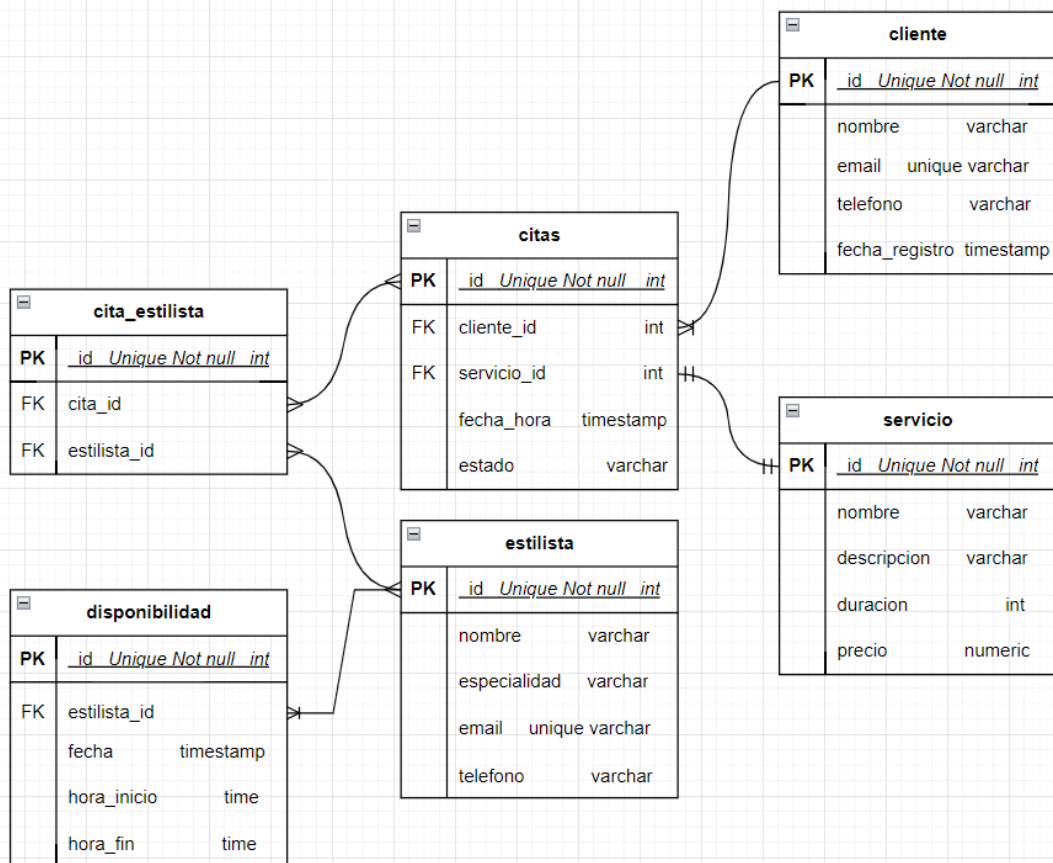
TEMÁTICA DE LA BASE DE DATOS

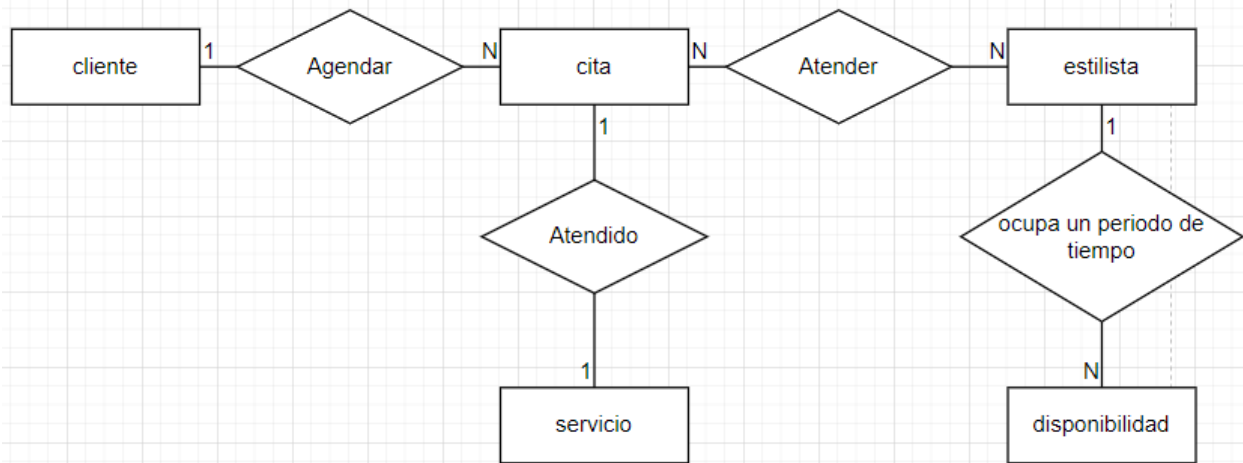
Corresponde a la base de datos creada en mariaDB de un sistema de agendamiento para una tienda que brinda servicio de estilista de uñas, la idea de negocio principal es una aplicación web mediante la cual los clientes de la tienda puedan registrarse, seleccionar el servicio que desean y agendar la cita correspondiente.

ENTIDADES

- Cliente
- Servicio
- Cita
- Estilista
- Disponibilidad

DIAGRAMAS DE ENTIDAD - RELACIÓN





LISTADO DE TABLAS

- cliente: Almacena la información de los clientes.
- servicio: Almacena los diferentes servicios que ofrece la tienda.
- cita: Almacena las citas agendadas por los clientes.
- estilista: Almacena la información de cada estilista.
- disponibilidad: Guarda los registros de disponibilidad para los estilistas es decir el periodo de tiempo en el cual el estilista estará ocupado.
- cita_Estilista: Tabla resultante de la relación N:M para asociar las citas con los estilistas.

DESCRIPCIÓN DE LOS CAMPOS DE CADA TABLA

- **Cliente:**
 - *id int primary key not null auto-increment*
 - *nombre varchar(100)*
 - *email varchar(100) unique*
 - *telefono varchar(15)*
 - *fecha_registro timestamp con la fecha y hora actual*
- **Servicio:**
 - *id int primary key not null auto-increment*
 - *nombre varchar(100)*
 - *descripcion varchar(255)*
 - *duracion int (en minutos)*
 - *precio numeric*
- **Cita:**
 - *id int primary key not null auto-increment*

- *cliente_id int foreign key de la tabla cliente(id)*
- *servicio_id int foreign key de la tabla servicio(id)*
- *fecha_hora timestamp con la fecha y hora*
- *estado varchar(20) para determinar si la cita está "pendiente", "confirmada", "cancelada"*
- **Estilista:**
 - *id int primary key not null auto-increment*
 - *nombre varchar(100)*
 - *especialidad varchar(100)*
 - *email varchar(100) unique*
 - *telefono varchar(15)*
- **Disponibilidad:**
 - *id int primary key not null auto-increment*
 - *estilista_id int foreign key de la tabla estilista(id)*
 - *fecha date*
 - *hora_inicio time*
 - *hora_fin time*
- **Cita_Estilista:**
 - *id int primary key not null auto-increment*
 - *cita_id int foreign key de la tabla cliente(id)*
 - *estilista_id int foreign key de la tabla estilista(id)*

VISTAS

- **vista_citas:**
 - **Propósito:** Proporciona una vista consolidada de todas las citas agendadas, incluyendo detalles del cliente, servicio, estilista, fecha y hora de la cita, y su estado.
 - **Columnas:**
 - *cita_id*: Identificador único de la cita.
 - *cliente_nombre*: Nombre del cliente que ha agendado la cita.
 - *cliente_email*: Correo electrónico del cliente.
 - *servicio_nombre*: Nombre del servicio solicitado.
 - *servicio_precio*: Precio del servicio.
 - *estilista_nombre*: Nombre del estilista asignado a la cita.
 - *fecha_hora*: Fecha y hora de la cita.
 - *estado*: Estado actual de la cita (por ejemplo, agendada, cancelada).

- **vista_disponibilidad_estilista:**
 - **Propósito:** Muestra la disponibilidad de cada estilista, incluyendo su nombre, especialidad, y los horarios disponibles.
 - **Columnas:**
 - *estilista_id*: Identificador único del estilista.
 - *estilista_nombre*: Nombre del estilista.
 - *especialidad*: Especialidad del estilista (por ejemplo, manicura, pedicura).
 - *fecha*: Fecha en la que el estilista está disponible.
 - *hora_inicio*: Hora de inicio de la disponibilidad del estilista.
 - *hora_fin*: Hora de fin de la disponibilidad del estilista.

Estas vistas son útiles para la gestión y visualización de citas y disponibilidad en la aplicación web, facilitando la interacción con los datos de manera más intuitiva y organizada.

FUNCIONES

- **obtener_numero_citas_estilista:**
 - **Propósito:** Esta función devuelve el número total de citas asignadas a un estilista específico.
 - **Parámetro:**
 - **estilista_id_param:** El identificador único del estilista para el cual se desea contar las citas.
 - **Funcionamiento:** La función realiza una consulta en la tabla *cita_estilista* para contar cuántas citas están asociadas con el estilista cuyo ID se pasa como parámetro. Devuelve este conteo como un número entero.
 - **Ejemplo de uso:**
SELECT obtener_numero_citas_estilista(2);
- **obtener_precio_total_cliente:**
 - **Propósito:** Esta función calcula el precio total que un cliente debe pagar por los servicios solicitados en sus citas.
 - **Parámetro:**
 - **cliente_id_param:** El identificador único del cliente para el cual se desea calcular el precio total.
 - **Funcionamiento:** La función suma los precios de todos los servicios asociados a las citas del cliente cuyo ID se pasa como parámetro. Realiza una consulta que une las tablas *cita* y *servicio* para obtener los precios de los servicios y devuelve la suma total como un valor numérico.
 - **Ejemplo de uso:**
SELECT obtener_precio_total_cliente(1);

Ambas funciones son útiles para obtener información específica sobre las citas y los costos asociados, facilitando la gestión de datos en el sistema de agendamiento.

STORED PROCEDURES

- **cancelar_cita:** recibe como parámetro el id de la cita que se quiere cancelar, marca el estado de esta cita en "cancelado" y borra los registros de la tabla **cita_estilista** y la correspondiente relación de tabla disponibilidad que tenga el id del estilista relacionado con la cita que se quiere cancelar.
 - **Eliminación de registros:**
 - Primero, se actualiza la tabla cita para cambiar el estado a 'cancelado'.
 - Después, se eliminan los registros relacionados en la tabla cita_estilista.
 - Finalmente, se eliminan los registros en disponibilidad que están relacionados con la cita y estilista.
- **obtener_citas_en_rango:** devuelve una tabla con las citas agendadas dentro de un rango de fechas especificado. Toma dos parámetros: *fecha_inicio* y *fecha_fin*, y devuelve una tabla con detalles de las citas, incluyendo el cliente, servicio, estilista, fecha y hora de la cita, y su estado.

Ejemplo de uso:

CALL obtener_citas_en_rango('2024-01-01 00:00:00', '2024-12-31 23:59:59');

Link del repositorio en github con el código SQL

https://github.com/Sergio20100/cursoSQL_coderhouse